AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method to assign tasks, comprising:

receiving a request to execute a task on one of a plurality of processors;

determining a task type for said task;

retrieving a processor task value of said task type for each processor, said processor task value representing a number of other task types affected by assigning said task to a processor;

selecting a processor from said plurality of processors based on said processor task values; and

assigning said task to said selected processor; and

updating said processor task values for each task type and each processor, wherein said updating comprises:

retrieving a first resource cost value for a first task type;

retrieving a second resource cost value for a second task type;

generating a modulo of said first resource cost value divided by said second resource cost value to form a first relative resource value for said first task type relative to said second task type;

determining a remaining resource value for said processor;

generating a modulo of said remaining resource value divided by said second resource cost value to form a second spare resource value for said second task type;

comparing said first relative resource value to said second spare resource value; and

updating said processor task value for said first task type in accordance with said comparison.

2.- 4. (Cancelled)

5. (Currently amended) The method of ~~claim 4~~claim 1, wherein said processor task value for said first task type is incremented by one if said first relative resource value is less than said second spare resource value.

6. (Currently amended) ~~The method of claim 1,~~ A method to assign tasks, comprising:

receiving a request to execute a task on one of a plurality of processors;

determining a task type for said task;

retrieving a processor task value of said task type for each processor, said processor task value representing a number of other task types affected by assigning said task to a processor;

selecting a processor from said plurality of processors based on said processor task values; and

assigning said task to said selected processor; and

updating said processor task values for each task type and each processor, wherein said updating comprises:

retrieving a first resource cost value for a first task type;

retrieving a second resource cost value for a second task type;

generating a modulo of said second resource cost value divided by said first resource cost value to form a second relative resource value for said second task type relative to said first task type;

determining a remaining resource value for said processor;

generating a modulo of said remaining resource value divided by said first resource cost value to form a first spare resource value for said first task type;

comparing said second relative resource value to said first spare resource value; and

updating said processor task value for said second task type in accordance with said comparison.

7. (Original) The method of claim 6, wherein said processor task value for said second task type is incremented by one if said second relative resource value is less than said first spare resource value.

8. (Currently amended) The method of claim 1, ~~wherein said updating comprises:~~

~~retrieving a first resource cost value for a first task type;~~

~~retrieving a second resource cost value for a second task type;~~

~~generating a modulo of said first resource cost value divided by said second resource cost value to form a first relative resource value for said first task type relative to said second task type;~~

~~generating a modulo of said second resource cost value divided by said first resource cost value to form a second relative resource value for said second task type relative to said first task type;~~

~~determining a remaining resource value for said processor;~~

~~generating a modulo of said remaining resource value divided by said first resource cost value to form a first spare resource value for said first task type;~~

~~generating a modulo of said remaining resource value divided by said second resource cost value to form a second spare resource value for said second task type;~~

~~incrementing said processor task value for said first task type if said first relative resource value is greater than said second spare resource value; and~~

~~incrementing~~ wherein said processor task value for said second task type is incremented if said second relative resource value is greater than said first spare resource value.

9. (Original) The method of claim 1, wherein said selecting comprises:

comparing said processor task values for said processors; and

selecting a processor having a highest processor task value.

10.- 15. (Cancelled)

16. (Currently amended) An article, comprising:

a storage medium;

said storage medium including stored instructions that, when executed by a processor, result in assigning tasks by receiving a request to execute a task on one of a plurality of processors, determining a task type for said task, retrieving a processor task value of said task type for each processor, selecting a processor from said plurality of processors based on said processor task values, assigning said task to said selected processor, said processor task value representing a number of other task types affected by assigning said task to said processor and updating said processor task values for each task type and each processor,

wherein the stored instructions, when executed by a processor, further result in said updating by retrieving a first resource cost value for a first task type, retrieving a second resource cost value for a second task type, generating a modulo of said first resource cost value divided by said second resource cost value to form a first relative resource value for said first task type relative to said second task type, determining a remaining resource value for said processor, generating a modulo of said remaining resource value divided by said second resource cost value to form a second spare resource value for said second task type, comparing said first relative resource value to said second spare resource value, and updating said processor task value for said first task type in accordance with said comparison.

17.-19. (Cancelled)

20. (Currently amended) The article of claim 16, wherein the stored instructions, when executed by a processor, further result in said updating by ~~retrieving a first resource cost value for a first task type, retrieving a second resource cost value for a second task type, generating a modulo of said first resource cost value divided by said second resource cost value to form a first relative resource value for said first task type relative to said second task type, generating a modulo of said second resource cost value divided by said first resource cost value to form a~~

~~second relative resource value for said second task type relative to said first task type, determining a remaining resource value for said processor, generating a modulo of said remaining resource value divided by said first resource cost value to form a first spare resource value for said first task type, generating a modulo of said remaining resource value divided by said second resource cost value to form a second spare resource value for said second task type, incrementing said processor task value for said first task type if said first relative resource value is greater than said second spare resource value, and~~ incrementing said processor task value for said second task type if said second relative resource value is greater than said first spare resource value.

21. (Previously presented) The article of claim 16, wherein the stored instructions, when executed by a processor, further result in said selecting by comparing said processor task values for said processors, and selecting a processor having a highest processor task value.